Security and Performance Analysis of Edge Computing in IoT

RADMEHR GHADIRI, University of Twente, The Netherlands

The Internet of Things (IoT) has become a ubiquitous part of our daily lives, with millions of sensors and devices gathering and exchanging data through complex networks. To address the problem of resource congestion, a new paradigm called edge computing has been introduced. Edge computing moves data computation and storage to the network edge, near the end-users, to offload computational stress from centralized data centers and reduce latency. However, this shift may introduce new cybersecurity threats and attacks that need to be addressed to ensure secure IoT deployment. Ensuring the security of edge computing is critical to maintaining the integrity and privacy of sensitive data and applications. In this study, we will simulate edge computing in IoT and perform experiments to analyze the performance and security of edge computing. By simulating IoT, we can create a controlled environment to test the system's capabilities and identify potential weaknesses. Based on our findings, we will offer recommendations for improving security and enhancing the performance of edge computing. Our aim is to provide practical insights that can help organizations deploy edge computing in a secure and efficient manner, while also mitigating potential risks and threats to IoT networks.

Additional Key Words and Phrases: Edge computing, Internet of Things, Security, Performance

1 INTRODUCTION

With the development of information technology, the Internet of Things (IoT) has become an important part of our daily lives [4]. Interconnected sensors and devices collect and exchange data through modern communication network infrastructure connected by millions of IoT nodes[1]. A variety of IoT applications can provide more accurate and fine-grained network services for users. However, the increasing number of sensors and devices interconnected via IoT techniques generates demands that conventional cloud computing-based services cannot satisfy. The computation processes need to be uploaded to the cloud, which is far from the end users, and limited bandwidth and network resources are occupied by massive data transmissions. This results in large network latency, which is unacceptable for time-sensitive IoT applications that could impact safety and emergency response[10].

Most IoT devices have limited power, and to extend their lifetime, it is necessary to balance power consumption by scheduling computation to devices with higher power and computational capabilities. In addition, processing data in computation nodes with the shortest distance to the user will reduce transmission time. Edge computing, which encompasses data computing and storage being performed at the network "edge" nearby the user, has been introduced to address these issues. Edge computing nodes are close to end-users, which can alleviate the peak in traffic flows and mitigate the bandwidth requirements of the centralized network, reducing the transmission latency during data computing or storage in IoT[6].

By distributing computation nodes deployed at the edge, traffic and computational pressure can be offloaded from the centralized cloud, and the response times of IoT applications can be faster than corresponding cloud computing services. Furthermore, edge computing can migrate computational and communication overhead from nodes with limited battery or power supply to edge nodes with significant power resources, thereby extending the lifetime of the nodes with limited battery and increasing the lifetime of the entire IoT network[10].

It is important to analyze the security and performance of edge computing in IoT because as more devices become interconnected through IoT, there is a greater risk of cyber attacks and security breaches [2] . Additionally, with the increasing demand for faster and more reliable network services, it is crucial to understand how edge computing can improve the performance of IoT applications. This will allow for the development of effective security measures and optimization strategies for edge computing in IoT.

This paper aims to answer the following research questions:

(1) What is the current state of research on Edge Computing for IoT, and how have researchers explored the use of Edge Computing in various applications?
(2) How does edge computing affect the overall performance of IoT systems?
(3) What are the key security risks associated with edge computing in IoT, and how can they be effectively mitigated?

The paper is structured as follows: Section 2 is the literature review, providing an overview of existing research and knowledge on the topic. Section 3 is the methodology, describing the research design, methods, and procedures. Section 4 illustrates the results of the experiments. Next, section 5 discusses the implications of the results of the conducted experiments. Finally, Section 6 provides a conclusion for this paper.

2 LITERATURE REVIEW

In recent years, there has been a surge in research regarding edge computing, despite its relatively new status.

One of the most influential studies in this field was conducted by Yu W et al.[10], which provides a comprehensive analysis of edge computing. The study explores various facets of edge computing, including its architecture, performance, and security. The authors also discuss the different types of edge devices and the ways in which they can be integrated into the network. Furthermore, the paper discusses the key challenges and future research directions for edge computing. The insights gained from this study have proven invaluable in shaping the development and adoption of edge computing technology.

In addition to Yu's study, Premsankar et al. [5] conducted a case study to investigate the suitability of edge computing for emerging IoT applications. The research involved the deployment of an edge

TScIT 39, July 7, 2023, Enschede, The Netherlands

computing infrastructure to support a smart city application. The study found that edge computing was highly suitable for the application, as it improved data processing efficiency and reduced network latency. The authors also highlighted the need for further research in the area of edge computing, particularly in the development of edge-based algorithms and applications.

Another paper [9] discusses the neglected aspect of security threats in edge computing platforms and their associated applications. The researchers provide a comprehensive survey focusing on influential and fundamental attacks specific to edge computing. The survey encompasses four main types of attacks: distributed denial of service (DDoS) attacks, side-channel attacks, malware injection attacks, and authentication and authorization attacks. These four attack types collectively account for 82% of the edge computing attacks reported by Statista. In addition to detailing the characteristics and implications of these attacks, the paper analyzes their root causes. It highlights the current state of edge computing security, and grand challenges that need to be addressed, and identifies future research directions in this field. Overall, the paper contributes to the understanding of security concerns in edge computing systems. It serves as a valuable resource for researchers and practitioners interested in mitigating the risks associated with edge computing deployments.

With regards to the security of edge computing, another key contribution to the field is the research conducted by Wang et al. [8], which analyzed the security of data collection models in edge computing. The authors identified the potential security risks associated with edge computing, such as data tampering and cyber attacks. They also proposed a secure data collection model that utilizes encryption and authentication techniques to protect sensitive data. The study has important implications for the development and adoption of edge computing, as security concerns are a key barrier to its widespread adoption.

Last but not least, Skirelis et al.[7] conducted a study to improve IoT solutions using Edge Computing. They aimed to optimize network structures and determine which parameters significantly affect system performance. The authors conducted simulations on various network topologies and different complexities in network bandwidth and delay parameters. Their findings suggest that the IoT configuration network is more sensitive to network topology, while the Internet configuration is more sensitive to network parameters. The study provides recommendations based on the results acquired to optimize IoT solutions using Edge Computing. Overall, the research provides valuable insights into optimizing IoT solutions for better performance.

With such valuable insights being generated through research efforts, it is evident that the significance of edge computing cannot be overstated, necessitating further research to tackle challenges and optimize its benefits as the field rapidly evolves, emphasizing the crucial collaboration between researchers and industry professionals to drive its continued development and widespread adoption.

3 RESEARCH METHODOLOGY

In this section, the measurement tools and environment of the performance and security analysis experiments will be described.

3.1 Performance Analysis

The setup included Mininet on Ubuntu 20.04.1 as the measurement environment, and the "ping," "top," and "/proc/meminfo" utilities as the primary measurement tools.

The measurement environment was established using Mininet, a network emulation tool running on Ubuntu 20.04.1. It provided a realistic and scalable environment for simulating edge computing scenarios and IoT devices. Three distinct topologies were considered: one edge device and three IoT devices (referred to as Topology1), one edge device and twenty IoT devices (referred to as Topology2), and two edge devices and twenty IoT devices (referred to as Topology3). These topologies represented different deployment scenarios encountered in edge computing.

The nodes were represented as lightweight virtual machines running on the host machine. Each node was allocated a single CPU core from the host machine, ensuring that the nodes had access to processing power for executing tasks and handling network traffic. Additionally, each node was allocated 981.2MB of memory, providing a limited but sufficient amount of memory for the processes running inside the nodes. These specifications struck a balance between resource usage and the ability to simulate and test network scenarios effectively.

The primary measurement tools used were the "ping" utility for network latency and connectivity, the "top" command for CPU usage, and the "/proc/meminfo" file for monitoring memory usage. Ping was utilized to measure the round-trip time (latency) for packet transmission and reception between the edge device and IoT devices. Additionally, the "top" command provided information on CPU usage, allowing for monitoring the CPU utilization of the system and individual processes. Memory usage was tracked by reading the contents of the "/proc/meminfo" file, which provided insights into memory utilization. These measurements collectively provided a comprehensive understanding of latency, throughput, memory usage, and CPU usage in the network scenarios tested.

In addition to latency and throughput measurements, memory usage and CPU usage were also considered. Memory usage was monitored by reading the contents of the "/proc/meminfo" file, which provided information about memory utilization. CPU usage information was extracted using the "top" command, which allowed for monitoring the CPU usage of the system and individual processes. These measurements provided insights into the resource utilization and performance of the network in different scenarios.

In the low workload scenarios, one packet of 64 KB was sent from the edge device to the IoT devices, while in the high workload scenarios, three packets of 500 KB each were sent. This variation in the number of packets allowed for evaluating the impact of the workload on latency, throughput, memory usage, and CPU usage more comprehensively. Each scenario was repeated 25 times to ensure statistical reliability and accuracy.

3.2 Security Analysis

In addition to the performance analysis, the research project followed a comprehensive methodology to assess the security of edge computing, utilizing the Mininet virtual network emulator on Ubuntu 20.04.1. The evaluation focused on four attack vectors: Unauthorized

Access, Data Tampering, Denial of Service (DoS), and Side Channel Attack.

To evaluate Unauthorized Access, an IoT device acted as the attacker, conducting a dictionary attack on the edge device(s). This involved systematically attempting passwords from a predefined list to gain unauthorized access.

Data Tampering was assessed through three methods. First, checksum verification was performed by processing a sample data string on the edge device(s) and computing a checksum using the MD5 hashing algorithm. The resulting checksum was compared to the expected value to ensure data integrity. Second, data tampering with digital signatures was tested using the RSA algorithm and last but not least, the SHA-256 hash function was used to detect tampering attempts.

The Denial of Service (DoS) scenario involved launching a flooding attack on the edge device(s) using the hping3 command. This flood attack inundated the target IP address with a high volume of UDP packets, evaluating the edge device(s)' resilience to such attacks and uncovering potential vulnerabilities.

Regarding the Side Channel Attack, a node was considered an attacker to try measuring the time taken to perform an operation on the victim. Investigating this attack vector will help identify vulnerabilities associated with side-channel attacks in edge computing systems.

Importantly, the simulated attack scenarios were conducted on the same network topologies as those used in the performance analysis section. These consistent topologies closely resembled real-world edge computing environments, ensuring the relevance and applicability of the findings to practical scenarios. This approach also allows for a comprehensive understanding of the interplay between performance and security in edge computing, enabling the development of holistic solutions that balance both aspects effectively.

4 RESULTS

In this section, the results of the experiments regarding performance and security analysis are presented.

4.1 Performance

The performance analysis of edge computing in IoT encompassed five aspects: latency, throughput, scalability, memory usage, and CPU usage. The results were presented through four distinct graphs, each capturing the findings for a specific aspect. Scalability is discussed in section 5. These graphs, accompanied by numerical values, provide a comprehensive overview of the performance evaluation conducted in this research.

Figure 1 illustrates the latency results obtained for different scenarios and topologies. It demonstrates the impact of workload and packet size on latency, with larger workloads and packet sizes leading to increased latency. It also illustrates how increasing edge devices can decrease latency. Among the three topologies, topology 1 consistently exhibits the lowest latency, while topology 2 demonstrates the highest latency. These findings underscore the importance of workload management and network configuration in minimizing latency in edge computing systems for IoT.

Figure 2 showcases the throughput results for the various scenarios and topologies. It depicts the relationship between packet size, workload, and the network's capacity to handle data transmission. As the packet size increases, the throughput increases. Topology 1 consistently achieves the highest throughput, while topology 2 exhibits relatively lower throughput. These results emphasize the significance of edge computing in optimizing network resources to enhance the throughput of edge computing systems.

Figure 3 presents the memory usage results for the different scenarios and topologies. It illustrates the relationship between packet size, workload, and the network's memory utilization. Overall, increasing the number of nodes or packet size only slightly increases memory usage. Topology 2 consistently demonstrates slightly higher memory usage indicating the additional resources required for supporting a larger number of devices. However, the impact on memory usage remains relatively small across all scenarios and topologies. These findings underscore the efficient utilization of memory resources in edge computing systems, as even with increased workload or device count, the memory requirements remain within manageable limits.

Figure 4 showcases the CPU usage results for the various scenarios and topologies. It illustrates the relationship between packet size, workload, and the network's CPU utilization. As the number of nodes or packet size increases, the CPU usage also increases. Topology 2 consistently exhibits higher CPU usage compared to Topology 1 and 3, indicating the additional processing requirements for managing a larger number of devices and handling increased data traffic. These findings emphasize the importance of considering CPU resources when designing and scaling edge computing systems. Efficient resource allocation and load balancing techniques can help optimize CPU usage and ensure the smooth operation of edge computing environments even under high workload conditions.

The availability of these four graphs, along with the corresponding numerical values, enables a comprehensive understanding of the performance analysis. The graphs provide visual representations of the trends and patterns, while the numerical values offer measurements for further analysis and comparison. Moreover, the mentioned results will help to determine the scalability of edge computing which will be elaborated on in section 5.

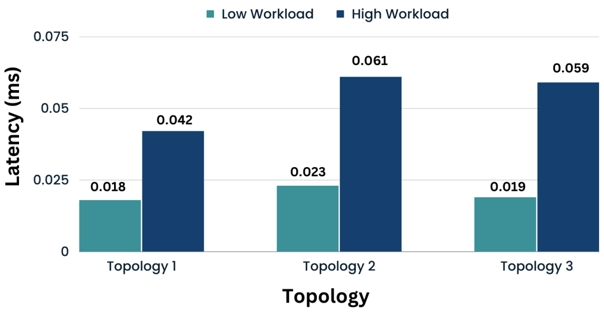

Fig. 1. Latency Performance in Different Topologies

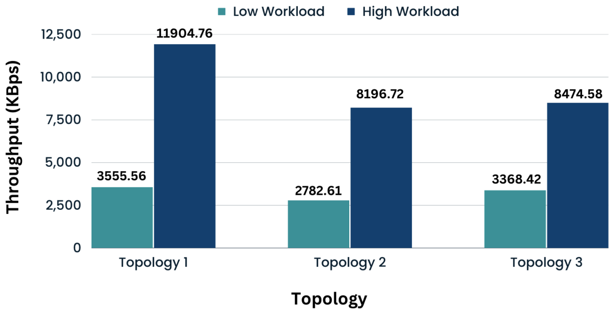

Fig. 2. Throughput Performance in Different Topologies

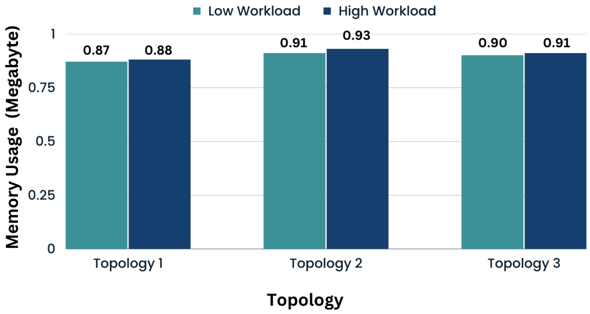

Fig. 3. Memory Usage in Different Topologies

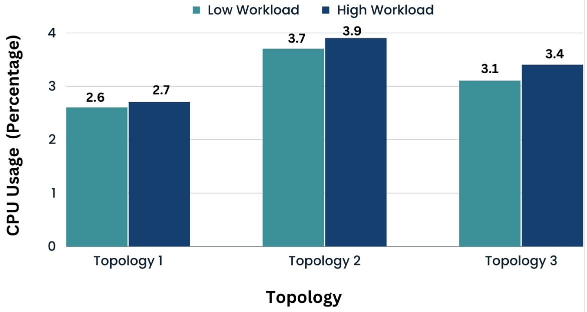

Fig. 4. CPU Usage in Different Topologies

4.2 Security

The results of the security performance experiments provide valuable insights into the security of edge computing systems. The simulations revealed important findings for each of the attack vectors considered: Unauthorized Access, Data Tampering, Denial of Service (DoS), and Side Channel Attack. It is important to note that with regard to security analysis, this paper focuses on the feasibility of each of the attack vectors and the quantitative analysis of these attacks is not in the scope of this paper.

In terms of Unauthorized Access, the dictionary attack simulations demonstrated the feasibility of gaining unauthorized access to the edge device(s). The experiments showed that the edge device(s) were prone to Unauthorized Access via dictionary attack due to lower security measures in comparison with cloud infrastructures. This highlights the critical importance of implementing strong passwords and robust access control mechanisms in edge devices to prevent unauthorized access.

The evaluation of Data Tampering methods, including checksum verification, digital signatures, and the SHA-256 hash function yielded positive results. The simulations showed that all the methods were effective in detecting tampering attempts and ensuring data integrity. The checksum verification using the MD5 hashing algorithm accurately identified any changes to the data by comparing the computed checksum with the expected value. Additionally, the digital signatures generated using the RSA algorithm and SHA-256 hash function successfully detected tampering attempts. These findings validate the reliability of these methods in protecting the integrity of data in edge computing environments.

In terms of Denial of Service (DoS), the simulations demonstrated the vulnerability of the edge device(s) to flooding attacks. The system was easily overwhelmed by the high volume of UDP packets, leading to unresponsiveness or complete inaccessibility. This emphasizes the critical need for robust DoS mitigation mechanisms to safeguard edge computing systems from such attacks and ensure uninterrupted operation.

Regarding the Side Channel Attack, the simulations demonstrated the possibility that the attacker measures the time taken to perform an operation on the victim. This matter might cause significant issues which will be discussed in section 5.

5 DISCUSSION

The experiments that were performed in this paper yielded noticeable results that will be discussed below.

5.1 Performance

The observed behavior of the system provides insights into its scalability characteristics. When increasing the number of nodes in the system, the marginal increase in latency suggests a favorable scalability profile. Scalability refers to a system's ability to accommodate a growing number of resources and handle increasing workloads while maintaining acceptable performance levels.

In our experiments, the minimal impact on latency despite the addition of more nodes indicates that the system can efficiently scale and distribute the workload among the nodes. This implies that the system can handle increased demand without significant degradation in performance.

Similarly, when scaling up the packet sizes and quantities, the slight increase in latency alongside an increase in throughput signifies a scalable architecture. The system can handle larger data payloads and a higher volume of packets while maintaining relatively stable latency levels. The improved throughput suggests that the system effectively utilizes the expanded resources to process and transmit data, accommodating the increased workload. Moreover, the addition of edge devices demonstrated a decrease in latency and an increase in throughput, indicating the successful offloading of processing tasks from the data center.

Overall, these findings indicate that the system exhibits favorable scalability characteristics. The ability to handle additional nodes, larger packet sizes, and increased workload while maintaining acceptable latency and achieving higher throughput demonstrates its capability to scale efficiently. This suggests that the system can accommodate growth and resource demands, making it suitable for scalable deployments in various scenarios.

The analysis of memory usage in this paper revealed intriguing findings. Notably, we observed that despite variations in node numbers or packet sizes, memory usage exhibited minimal changes. This stability in memory consumption indicates the efficiency of edge computing in managing memory resources within the IoT environment. By effectively processing and storing data within the limited memory capacities of edge devices, our findings suggest that edge computing can handle larger workloads without significant memory overhead. Moreover, the addition of edge devices demonstrated a decrease in memory usage, indicating successful offloading of processing tasks from the data center. This strategy not only optimizes memory utilization but also enhances scalability and flexibility, allowing the edge infrastructure to adapt to increasing demands without compromising memory performance.

Furthermore, the investigation into CPU usage during the experiments revealed interesting trends. As expected, CPU usage increased with higher node numbers and larger packet sizes, indicating the computational demands imposed by these factors. However, it is worth noting that the observed increase in CPU usage was relatively modest and manageable. This finding suggests that the edge computing framework effectively handles the computational requirements without overwhelming the available CPU resources. Moreover, the introduction of edge devices into the architecture resulted in a decrease in CPU usage. This reduction can be attributed to the offloading of processing tasks to the edge, thereby relieving the centralized cloud or data center from excessive computational burden. The ability of edge computing to optimize CPU usage while accommodating varying workloads highlights its potential for efficient resource utilization and scalability in IoT deployments.

5.2 Security

In terms of Unauthorized Access, our simulations using dictionary attacks revealed concerning results regarding the vulnerability of edge devices. The experiments demonstrated the feasibility of gaining unauthorized access to the edge device(s) due to lower security measures in comparison with cloud infrastructures. This finding highlights the critical importance of implementing strong passwords and robust access control mechanisms in edge devices to prevent unauthorized access. The susceptibility of edge devices to dictionary attacks emphasizes the need for enhanced security measures at the edge, considering that compromised edge devices can serve as gateways to critical systems and data within the IoT ecosystem. Implementing stringent password policies, such as enforcing complex and unique passwords, along with multi-factor authentication and role-based access controls, can significantly mitigate the risk of unauthorized access and bolster the overall security posture of edge computing environments.

The evaluation of Data Tampering methods, including checksum verification, digital signatures, and the SHA-256 hash function, yielded positive results in the research project. The simulations conducted demonstrated the effectiveness of these methods in detecting tampering attempts and ensuring data integrity within edge computing environments. The checksum verification, utilizing the MD5 hashing algorithm, proved to be a reliable technique by accurately identifying any changes to the data. This was achieved by comparing the computed checksum with the expected value, thereby providing a robust mechanism for data integrity. Furthermore, the digital signatures generated using the RSA algorithm and SHA-256 hash function successfully detected tampering attempts by providing strong cryptographic proof of authenticity. These findings validate the reliability and efficacy of these methods in protecting the integrity of data within edge computing architectures. By implementing these techniques, edge devices can mitigate the risks associated with data tampering, ensuring the trustworthiness of data and enhancing the overall security posture of the IoT ecosystem.

In terms of Denial of Service (DoS), the simulations conducted in the research project revealed the vulnerability of the edge device(s) to flooding attacks. The system's responsiveness was significantly impacted as it became easily overwhelmed by the high volume of UDP packets, leading to unresponsiveness or complete inaccessibility. These results underscore the critical importance of implementing robust DoS mitigation mechanisms to safeguard edge computing systems from such attacks and ensure uninterrupted operation. Effective DoS mitigation strategies may include rate limiting, traffic filtering, and anomaly detection techniques to identify and mitigate excessive or malicious network traffic [3]. Additionally, the implementation of intrusion detection and prevention systems at the edge can play a vital role in detecting and mitigating DoS attacks in real time. By proactively addressing DoS vulnerabilities and adopting resilient mitigation measures, edge computing environments can enhance their overall availability and reliability, safeguarding critical IoT services from disruptions.

In regards to Side Channel Attacks, the simulations unequivocally demonstrated the feasibility for an attacker to exploit the time taken to perform operations on the victim. The results revealed that an adversary can leverage these time measurements to infer sensitive information, such as password guesses. By analyzing the timing variations, an attacker can deduce patterns and potentially break into the system by guessing passwords. These findings emphasize the critical importance of implementing robust countermeasures to mitigate the risks associated with side-channel attacks. Protecting against such attacks requires a multi-faceted approach that includes secure coding practices, implementing proper input validation, and applying techniques such as masking, blinding, or randomization to obfuscate sensitive timing information. By recognizing the vulnerabilities inherent in side-channel attacks, system designers and developers can enhance the security of edge computing environments and protect against potential breaches of sensitive information.

6 CONCLUSION

In conclusion, our research focused on the security and performance analysis of edge computing in the context of IoT. Through our investigations, we found that edge computing offers significant advantages in terms of performance and scalability. It demonstrated commendable performance even with increasing node numbers and packet sizes, while the addition of edge devices helped reduce resource consumption. These findings highlight the potential of edge computing to efficiently handle varying workloads and alleviate the burden on centralized cloud infrastructures.

However, our research also uncovered the inherent security challenges associated with edge computing. The simulations revealed vulnerabilities to unauthorized access and side-channel attacks, emphasizing the need for robust access control mechanisms and countermeasures against timing-based attacks.

Our findings underscore the importance of addressing the security concerns in edge computing. It is essential to develop and implement comprehensive security measures, including strong authentication, encryption, and intrusion detection systems, to safeguard edge devices and protect the sensitive data they handle. Furthermore, ongoing research is crucial to further enhance the security and privacy aspects of edge computing in the evolving landscape of IoT.

Overall, while edge computing exhibits promising performance benefits, its security vulnerabilities necessitate a comprehensive and proactive approach to ensure the confidentiality, integrity, and availability of data and services. Future research endeavors should focus on developing advanced security mechanisms tailored specifically for edge computing environments, as well as evaluating their effectiveness in real-world deployments. By addressing these challenges, we can unlock the full potential of edge computing in IoT while ensuring the utmost security and privacy for users and their data.

REFERENCES

[1] Mohammed El-Haii, Maroun Chamoun, Ahmad Fadlallah, and Ahmed Serhrouchni. 2019. Analysis of Cryptographic Algorithms on IoT Hardware platforms. *2018 2nd Cyber Security in Networking Conference, CSNet 2018* (1 2019). https://doi.org/10.1109/CSNET.2018.8602942

[2] Mohammed El-Hajj, Maroun Chamoun, Ahmad Fadlallah, and Ahmed Serhrouchni. 2017. Analysis of authentication techniques in Internet of Things (IoT). *2017 1st Cyber Security in Networking Conference, CSNet 2017* 2017-January (12 2017), 1–3. https://doi.org/10.1109/CSNET.2017.8242006

[3] Mohammed El-Hajj, Maroun Chamoun, Ahmad Fadlallah, and Ahmed Serhrouchni. 2018. Taxonomy of authentication techniques in Internet of Things (IoT). *IEEE Student Conference on Research and Development: Inspiring Technology for Humanity, SCOReD 2017 - Proceedings* 2018-January (2 2018), 67–71. https://doi.org/10.1109/SCORED.2017.8305419

[4] Mohammed El-Hajj, Ahmad Fadlallah, Maroun Chamoun, and Ahmed Serhrouchni. 2019. A survey of internet of things (IoT) authentication schemes. *Sensors* 19, 5 (2019), 1141.

[5] Gopika Premsankar, Mario Di Francesco, and Tarik Taleb. 2018. Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal* 5 (4 2018), 1275–1284. Issue 2. https://doi.org/10.1109/JIOT.2018.2805263

[6] Ju Ren, Pan Yi, Goscinski Andrzej, and Raheem A Beyah. 2018. Edge_Computing_for_the_Internet_of_Things. *IEEE Network* (2018).

[7] Julius Skirelis and Dalius Navakauskas. 2020. Performance analysis of edge computing in IoT. *Elektronika ir Elektrotechnika* 26 (2 2020), 72–77. Issue 1. https://doi.org/10.5755/j01.eie.26.1.23235

[8] Tian Wang, Lei Qiu, Arun Kumar Sangaiah, Anfeng Liu, Md Zakirul Alam Bhuiyan, and Ying Ma. 2020. Edge-Computing-Based Trustworthy Data Collection Model in the Internet of Things. *IEEE Internet of Things Journal* 7 (5 2020), 4218–4227. Issue 5. https://doi.org/10.1109/JIOT.2020.2966870

[9] Yinhao Xiao, Yizhen Jia, Chunchi Liu, Xiuzhen Cheng, Jiguo Yu, and Weifeng Lv. 2019. Edge Computing Security: State of the Art and Challenges. *Proc. IEEE* (2019). https://doi.org/10.1109/JPROC.2019.2918437

[10] Wei Yu, Fan Liang, Xiaofei He, William Grant Hatcher, Chao Lu, Jie Lin, and Xinyu Yang. 2017. A Survey on the Edge Computing for the Internet of Things. *IEEE Access* 6 (11 2017), 6900–6919. https://doi.org/10.1109/ACCESS.2017.2778504